UCRL-ID-132586

Shock Transmission and Reflection from a Material Interface and Subsequent Reflection from a Hard Boundary

O. A. Hurricane
P. L. Miller

November 20, 1998

Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract W-7405-ENG-48.

Shock Transmission and Reflection from a Material Interface and Subsequent Reflection from a Hard Boundary

O. A. Hurricane and P. L. Miller
Lawrence Livermore National Laboratory, University of California
P.O. Box 808, Livermore, CA 94550

As a shock wave passes through a material interface into a region of higher density (the receiver material), a transmitted and reflected shock wave are both generated and the interface is set into motion. The speeds of the transmitted shock, reflected shock, and interface are related to the initial shock speed and material properties via a set of coupled nonlinear equations that, in general, cannot be easily solved analytically. In this report, we derive the equations which describe this process and we document a numerical routine which solves the nonlinear equations. We then go on to solve the problem of finding the position where the interface collides with the transmitted shock wave once the transmitted shock wave is reflected from an impenetrable boundary located somewhere away from the initial material interface. Finally, we compare the analytical predictions with the CALE simulation running in 1-D.

I. INTRODUCTION

The problem of how a shock wave is transmitted through a material interface, subsequently reflects from a hard boundary, and then collides with the original material interface, sounds like a classic textbook problem. Given the age of the theory of one dimensional (1-D) shocks (e.g. Ref [1] and references therein) it comes as something of a surprise that the analytical solution of this problem is not readily available in the classic literature on the subject. In spite of the fact that the solution of this problem is only of textbook difficulty, it is nevertheless useful to provide a reference on the analytic/numerical solution of such a problem. This problem is particularly relevant for those involved in studying Richtmyer-Meshkov instabilities [2,3] in shock tubes (e.g Ref. [4]) or high power laser driven experiments.

In this report our problem is solved in two distinct parts. Section II describes the solution for the problem of how a shock incident on a material interface is transmitted, and partially reflected, from the interface and how this process sets the interface into motion. The equations that describe the transmission/reflection problem form a coupled nonlinear set which, in general, have no simple explicit solution but can instead be solved by numerical techniques. Section III deals with the problem of finding the position where the transmitted shock, upon later reflection from a impenetrable wall, collides with the moving material interface. In Section IV, predictions of the analytical formulation are compared with results from running the CALE simulation in 1-D. Section V has some closing remarks. In Appendix A, we provide a fortran computer code which solves all of the equations presented in the body of this report. Throughout, we make the perfect gas assumption.

II. ANALYSIS OF THE SHOCK TRANSMISSION/REFLECTION PROBLEM

Our problem begins by supposing that we have two materials at rest in contact across a 1-D interface. We call the region to the left of the interface Region 0 and the region to the right of the interface Region 2. We assume that both Regions 0 and 2 are perfect gases with known macroscopic properties such as the density (ρ), pressure (p), sound speed (c), and polytropic index (γ). Figure 1 illustrates the problem setup. Initially, the pressures in Region 0 and Region 2 are assumed to be equal ($p_0 = p_2$) so that the interface is in stationary equilibrium.

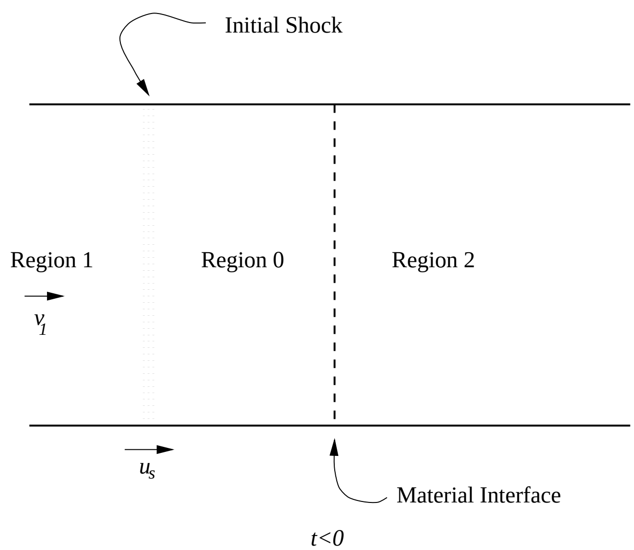

Fig. 1 A shock wave is traveling from left to right with a known speed u_s. As the shock moves through the material of Region 0 it processes the material into Region 1 material which has the same molecular properties as Region 0 material, but different thermodynamic properties, i.e. pressure, density, temperature, etc. At time $t = 0$ the shock will reach the boundary between Region 0 and Region 2.

We now suppose that we have a shock wave traveling from Region 0 toward Region 2 with a known Mach number, $M = u_s/c_0$, where u_s is the shock speed in Region 0 and c_0 is the sound speed in Region 0. (In what follows it is to be understood that a numerical subscript, i, on a quantity indicates the region, i, with which that quantity is associated). As the shock wave propagates through Region 0 the material is "processed" into a material with the same polytropic index, but with a different pressure, density, temperature, and internal energy. We call the region of material processed by the initial shock Region 1. Once the shock contacts the Region 0-Region 2 interface all of Region 0 material will have been processed into Region 1 material. From the Hugoniot relations (e.g. Ref. [1]), it is straight forward algebra to relate Region 1 material properties to Region 0 material properties in terms of the Mach number of the initial shock, i.e.

$$\frac{v_1}{c_0} = \frac{2}{\gamma_0+1}\frac{M^2-1}{M} \tag{1}$$

$$\frac{p_1-p_0}{p_0} = \frac{2\gamma_0}{\gamma_0+1}(M^2-1) \tag{2}$$

$$\rho_0\left(\frac{1}{\rho_1}-\frac{1}{\rho_0}\right) = \frac{2}{\gamma_0+1}\left(\frac{1}{M^2}-1\right) \tag{3}$$

where v_1 is the "piston velocity" driving the shock wave to the right. In the lab frame, the processed material behind the shock (Region 1) is flowing to the right with speed v_1. From Eq. (2) it is easy to show that $p_1 > p_2$ as long as $M > 1$ (with $p_0 = p_2$).

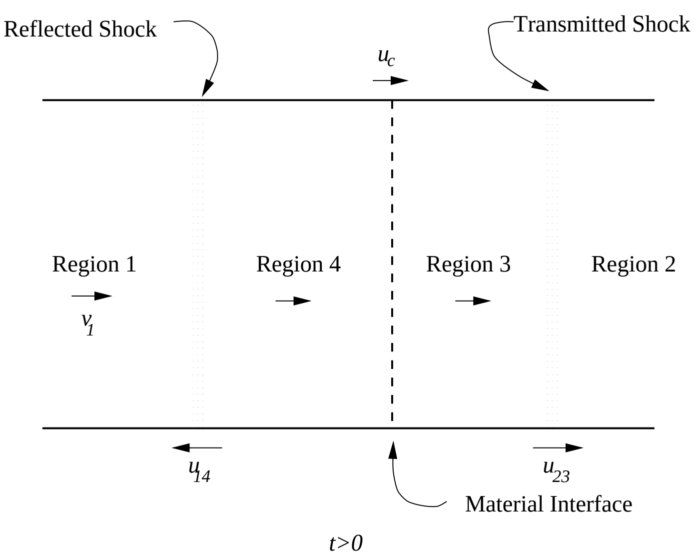

Fig. 2 Once the initial shock encounters the material interface between Region 0 and Region 2 a transmitted and reflected shock are produced and the interface is set into motion. The transmitted shock processes Region 2 material into Region 3 material while the reflected shock processes Region 1 material (formerly Region 0 material) into Region 4 material. All of the material between the reflected and transmitted shock moves with the speed of the interface, u_c.

Now assume that the density of Region 2 is greater than that of Region 0, $\rho_2 > \rho_0$. Once the initial shock wave encounters the material interface between Region 0 (now Region 1) and Region 2 a transmitted and reflected shock (as opposed to a rarefaction) are produced at the interface and the interface is set into motion (to the right) with speed, u_c (see Figure 2).

As the transmitted shock moves to the right in Region 2 it processes Region 2 material into what we call Region 3 material, maintaining the same polytropic index but changing the pressure and density. The Mach number of the transmitted shock is $M_{23} = u_{23}/c_2$, where u_{23} is the speed of the shock, in the lab frame, moving into Region 2. Similarly, the shock reflected from the interface processes Region 1 material into what we call Region 4 material and travels to the left with a Mach number $M_{14} = (u_{14}+v_1)/c_1$, where u_{14} is the speed of the shock in the lab frame. Note that since the material in Region 1 is flowing to the right with speed v_1 the Mach number M_{14} is not a lab frame quantity.

There are two key physical insights which allow solution of transmission/reflection part of this problem. First, the speed of the materials in Regions 3 and 4 are the same and equal to the speed of the interface, u_c. If the speeds of Region 3 and 4 where not the same the mass in the neighborhood of the interface would change in time. Second, across the interface the pressures must be equal ($p_3 = p_4$) so that the interface does not accelerate. Thus, u_c can be treated as an "effective" piston velocity driving the transmitted shock into Region 2 and $v_1 - u_c$ can be treated as the "effective" piston velocity driving the reflected shock into Region 1. Therefore, in a fashion similar to Eqs. (1)-(2) we have

$$\frac{u_c}{c_2} = \frac{2}{\gamma_2+1}\frac{M_{23}^2-1}{M_{23}} \tag{4}$$

$$\frac{p_c-p_2}{p_2} = \frac{2\gamma_2}{\gamma_2+1}(M_{23}^2-1) \tag{5}$$

$$\frac{v_1-u_c}{c_1} = \frac{2}{\gamma_0+1}\frac{M_{14}^2-1}{M_{14}} \tag{6}$$

$$\frac{p_c-p_1}{p_1} = \frac{2\gamma_0}{\gamma_0+1}(M_{14}^2-1) \tag{7}$$

where $p_c = p_3 = p_4$ is the pressure at the interface. These four coupled equations form a nonlinear system the solution of which yields the unknowns u_c, p_c, M_{23}, and M_{14}. In general, solution of this system is not practical analytically, so numerical techniques must be employed (see Appendix A). By inspection of Eqs. (4)-(7) we see that the necessary conditions for the production of both reflected and transmitted shocks are $p_c > p_1 > p_2$, and $v_1 > u_c > 0$.

III. ANALYSIS OF THE COLLISION OF THE INTERFACE AND WALL REFLECTED SHOCK

Once the initial shock is transmitted through the interface and the interface is set in motion, both move to the right at their respective speeds. We now suppose that the right hand side of our system is bounded by a wall. As long as $\gamma_2 > 1$ the shock moves faster than the interface, so it will encounter the wall and reflect from the wall before the interface has a chance to reach the wall itself. To calculate the position at which the wall reflected shock encounters the interface it is necessary to compute the speed of the shock upon reflection from the wall (this speed is different than the speed of the shock before reflection from the wall). Figure 3 summarizes that problem at this stage.

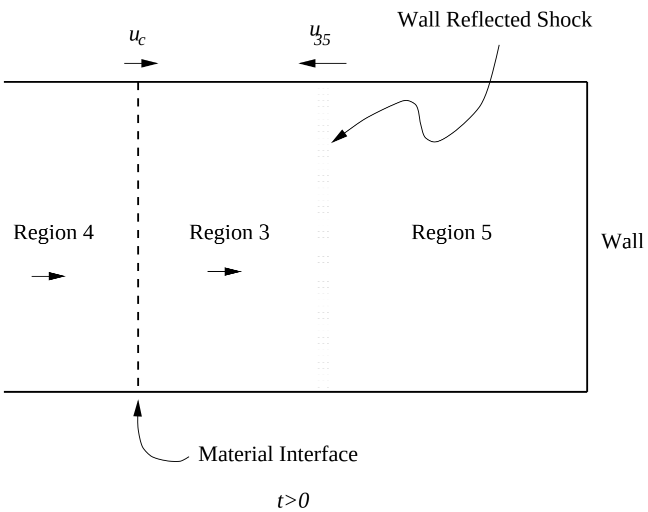

Fig. 3 The transmitted shock which processes Region 2 material into Region 3 material encounters a wall and reflects from it. The wall reflected shock then processes Region 3 material into Region 5 material bringing the material to rest. A short time later, the wall reflected shock will collide with the original material interface.

To compute the speed of the shock upon reflection from the wall it is necessary to realize that the material bounded by the reflected shock and the wall (Region 5) must be at rest to be consistent with the boundary condition at the wall. Thus, a relation similar to Eq. (1) is again valid

$$\frac{u_c}{c_3} = \frac{2}{\gamma_2 + 1}\frac{M_{35}^2 - 1}{M_{35}} \tag{8}$$

where $M_{35} = (u_{35}+u_c)/c_3$ is the Mach number of the wall reflected shock in the frame moving with the interface speed and u_{35} is the speed (in the lab frame) of the wall reflected shock. The sound speed $c_3 = \sqrt{\gamma_2 p_c/\rho_3}$ can be computed using a relation similar to Eq. (3), i.e.

$$\rho_2\left(\frac{1}{\rho_3} - \frac{1}{\rho_2}\right) = \frac{2}{\gamma_2 + 1}\left(\frac{1}{M_{23}^2} - 1\right). \tag{9}$$

If we assume that the undisturbed interface is originally at a distance L from the wall on the right side of the system, then simple mechanics (see Figure 4) tells us the position (x) of the collision between the interface and the wall reflected shock,

$$\frac{x}{L} = \frac{\frac{u_c}{u_{23}} + \frac{u_c}{u_{35}}}{1 + \frac{u_c}{u_{35}}}. \tag{10}$$

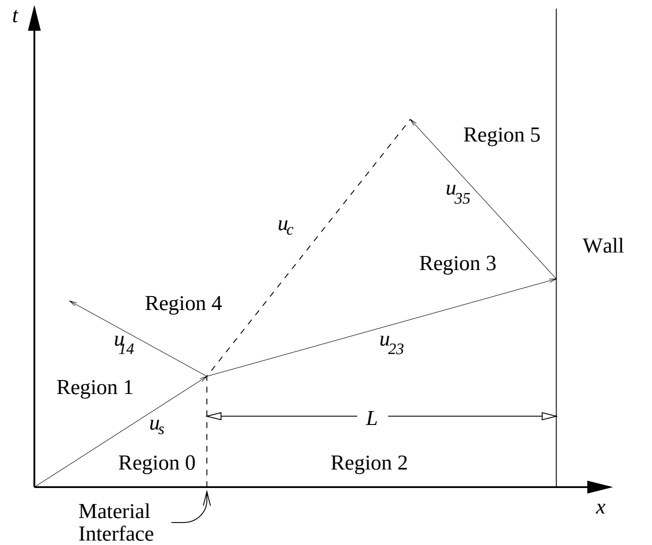

Fig. 4 The dynamical problem is easily visualized using a $x - t$ diagram. If the material interface is set into motion at $t = 0$ then the interface covers a distance $x = u_c T$ in time T. The time T is equal to the time it takes the transmitted shock to cover a distance L plus the time it takes the wall reflected shock to cover a distance $L - x$, i.e. $T = L/u_{23} + (L - x)/u_{35}$.

IV. COMPARISON WITH THE CALE SIMULATION

In Figure 5 we show how prediction of the position x from the above analysis compares with that predicted by the CALE simulation for a range of initial shock Mach numbers ranging from $M = 1.5$ to $M = 10$.

To produce Figure 5 the initial conditions in CALE are prepared in a way similar to the situation shown in Figure 1. We choose the material in Region 0 to be air with molecular weight 28.8 ($\rho_0 = 2.66 \times 10^{-4}\ g/cm^3$, $\gamma_0 = 1.4$, $c_0 = 0.0348\ cm/\mu s$, and $\varepsilon_0 = p_0/[(\gamma - 1)\rho_0] = 2.16 \times 10^{-3}\ Mbar \cdot cm^3/g$). The material in Region 2 is chosen to be SF_6 with molecular weight 142.1 ($\rho_2 = 1.31 \times 10^{-3}\ g/cm^3$, $\gamma_2 = 1.2$, and $\varepsilon_2 = 8.78 \times 10^{-4}\ Mbar \cdot cm^3/g$). CALE is simulating a shock tube of total length 472 cm with 680 grids in 1-D. The interface between Region 0 and Region 2 is initially set at 62 cm from the wall on the right hand side. In CALE, the ideal gas law equations of state are used for both materials. The simulation is run in pure Lagrangian mode.

V. CONCLUSION

As a shock wave passes through a material interface into a region of higher density (the receiver material), a transmitted and reflected shock wave are generated and the interface is set into motion. The speeds of the transmitted shock, reflected shock, and interface are related to the initial shock speed and material properties via a set of coupled nonlinear equations, Eqs. (4)-(7). Eqs. (4)-(7) cannot be solved explicitly, in general, but they can instead be solved using numerical techniques.

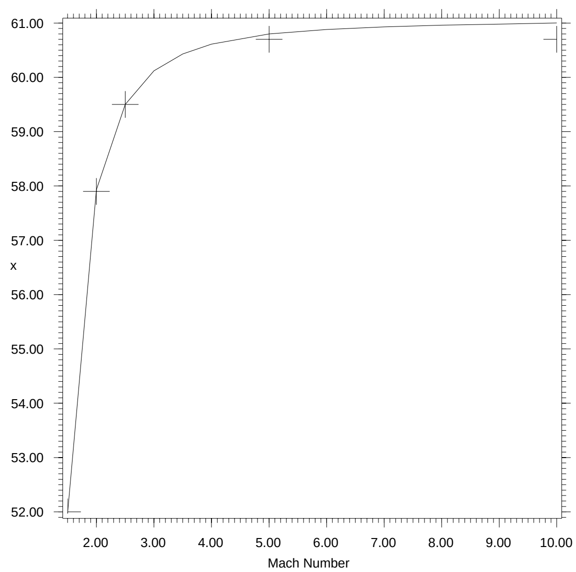

Fig. 5 Prediction of the point of collision (x in cm) between the interface and the wall reflected shock as a function of Mach number. The smooth curve is the analytical prediction and the crosses are the result obtained from the CALE simulation. Note that the zero is suppressed in this figure. The analytical and simulation results agree to within one percent.

Once the transmitted shock wave is reflected from a boundary located somewhere away from the initial material interface it will eventually collide with the material interface which was set into motion by the passage of the original shock wave. The analytical predictions, which come from numerical solution of Eqs. (1)-(10), compare quite well with results of running the CALE simulation in 1-D.

ACKNOWLEDGMENTS

This work was performed under the auspices of the United States Department of Energy by the Lawrence Livermore National Laboratory under Contract No. W-7405-ENG-48. The first author would like to acknowledge useful conversations with Dr. John Edwards on the topic of reference frames. Also, we would like to thank Dr. Peter Stry who suggested this problem. Those who wish to obtain an electronic copy of the *SHOCKINT* code may contact the first author of this report at hurricane1@llnl.gov.

APPENDIX A: FORTRAN CODE

This Appendix presents the numerical code *SHOCKINT* used to solve the coupled set of nonlinear equations, Eqs. (4)-(7). In addition, the code has also been set up to solve Eqs. (1)-(3) and Eqs. (8)-(10) so as to yield the full solution to the problem discussed in the body of this report. The numerical method used is a globally convergent Newton's method with line searching and backtracking [5]. The subroutines *newt*, *fdjac*, *fmin*, *lnsrch*, *lubksb*, and *ludcmp* are similar to those same routine found in Ref. [5]. Eqs. (4)-(7) are coded in the subroutine *funcv*.

The code uses an input file *shint.inp* to read in the material properties of Region 0 and Region 2 as well as the Mach number of the incident shock. An example of the contents of the *shint.inp* input file is shown below:

```
Sound speed (c0) for the undisturbed region (usually 0.0348 cm/us for air):
0.0348
Polytropic index (gamma) for region 0 (1.4 for air):
1.4
Density (r0) of the gas in region 0 (g/cm^3):
2.66e-4
Material energy (e0) of the gas in region 0 (Mbar-cm^3/g):
2.16165e-3
The Mach number (M) for the initial shock:
2.0
Polytropic index (gam2) for region 2 (1.2 for SF6):
1.2
Density (r2) of the gas in region 2 (g/cm^3):
1.31e-3
Material energy (e2) of the gas in region 2 (Mbar-cm^3/g):
8.7786e-4
```

An example of the output of the code *SHOCKINT*, for the input shown above, is given below:

```
 Input parameters:

 c0=  3.4800000E-02
 gamma=   1.400000
 rho0=  2.6599999E-04
 e0=  2.1616500E-03
 p0=  2.3009760E-07
 Mach #   2.000000
 gam2=   1.200000
 rho2=  1.3100000E-03
 e2=  8.7786000E-04
 p2=  2.3009760E-07
 Sound speed in region 2, c2=  1.4518142E-02

 Shock processed region 0 (i.e. region 1):

 p1/p0=   4.500000
 rho1/rho0=   2.666667
 rho1=  7.0933334E-04
 Piston Velocity, v1=  4.3499999E-02
 e1/e0=   1.687500
 e1=  3.6477842E-03
 Pressure boundary condition, p1=  1.0354391E-06
 Sound speed in region 1, c1=  4.5206524E-02

 Normal Return
 Eqn. 1  0.0000000E+00
 Eqn. 2 -2.9802322E-08
 Eqn. 3  0.0000000E+00
 Eqn. 4  5.9604645E-08

 Pressure at interface, pc=  1.6185573E-06
 Density in region 3, rho3=  5.6932378E-03

 Speeds of material interface and transmitted shock:

 Speed of the interface, uc=  2.8565962E-02
 Speed of shock transmitted into region 2, u23=  3.7103351E-02
 Speed of shock reflected from wall, u35=  1.1393987E-02
 Collision point, x/L=  0.9343911
```

Listed here is the fortran code for the *SHOCKINT* program. All subroutines necessary to compile the code are included.

```
C     shockint.f
C*******************************************************
C     This simple code computes the startup conditions for
C     the region "behind" the shock for use in the CALE
C     simulation.  Also, it computes the speed of the two-
C     material interface after contact with the initial
C     shock, the transmitted shock speed, and the speed
C     of the transmitted shock after hitting the "back" wall
C     of the shock tube.  Finally, it computes the position
C     of the point of collision between the wall reflected
C     shock and the material interface.
C
C     Units: lengths are in cm, times are in micro-sec.,
C            pressures are in Mega-bars, material
C            internal energies are in Mbar-cm^3/g, and
C            densities are in g/cm^3.
C
C     Oct. 20, 1998
C
C     Dr. Omar A. Hurricane
C     A-Division
C     LLNL
C     P.O. Box 808, L-312
C     Livermore, CA 91550
C
C     hurricane1@llnl.gov
C
C*********************************************************************
C
C                             NOTICE 1
C
C TO BE USED WHEN UC WILL NOT EXERT COPYRIGHT, BUT WHERE NOTIFICATION OF
C AN INTEREST TO COMMERCIALIZE IS DESIRABLE FOR VARIOUS REASONS.
C
C
C This work was produced at the University of California, Lawrence
C Livermore National Laboratory (UC LLNL) under contract no.
C W-7405-ENG-48 (Contract 48) between the U.S. Department of Energy (DOE)
C and The Regents of the University of California (University) for the
C operation of UC LLNL. The rights of the Federal Government are reserved
C under Contract 48 subject to the restrictions agreed upon by the DOE
C and University as allowed under DOE Acquisition Letter 97-1.
C
C
C DISCLAIMER
C
C
C This work was prepared as an account of work sponsored by an agency of
C the United States Government. Neither the United States Government nor
C the University of California nor any of their employees, makes any
C warranty, express or implied, or assumes any liability or
C responsibility for the accuracy, completeness, or usefulness of any
C information, apparatus, product, or process disclosed, or represents
C that its use would not infringe privately-owned rights.  Reference
C herein to any specific commercial products, process, or service by
C trade name, trademark, manufacturer or otherwise does not necessarily
C constitute or imply its endorsement, recommendation, or favoring by the
C United States Government or the University of California. The views and
C opinions of authors expressed herein do not necessarily state or
C reflect those of the United States Government or the University of
C California, and shall not be used for advertising or product
C endorsement purposes.
C
C
C NOTIFICATION OF COMMERCIAL USE
C
C
```

```
C Commericialization of this product is prohibited without notifying the
C Department of Energy (DOE) or Lawrence Livermore National Laboratory
C (LLNL).
C
C
C*******************************************************
      implicit none
      real p1op0,gamma,M,c0,r1or0,v1,e1oe0,pbc,r0,p0,e0
      real p1,r1,c1,gam2,r2,p2,e2,M23,M14,uc,pc,c2
      real r3,c3,M35,u23,u35,dum,x(4),f(4)
      logical check
      common /pars/ gamma,gam2,p2,p1,c1,c2,v1
      open (unit=10,file='shint.inp')
      rewind(10)
      read(10,*)
      read(10,*) c0      !Sound speed of Region 0 material.
      read(10,*)
      read(10,*) gamma   !Polytropic index of Region 0 material.
      read(10,*)
      read(10,*) r0      !Density of Region 0 material.
      read(10,*)
      read(10,*) e0      !Tmat of Region 0 material.
      read(10,*)
      read(10,*) M       !Mach # of initial shock.
      read(10,*)
      read(10,*) gam2    !Polytropic index of Region 2 material.
      read(10,*)
      read(10,*) r2      !Density of Region 2 material.
      read(10,*)
      read(10,*) e2      !Temp. of Region 2 material.
      close(10)
      write(*,*) 'Input parameters:'
      write(*,*)
      write(*,*) 'c0=',c0
      write(*,*) 'gamma=',gamma
      write(*,*) 'rho0=',r0
      write(*,*) 'e0=',e0
      p0=r0*c0*c0/gamma
      write(*,*) 'p0=',p0
      write(*,*) 'Mach #',M
      write(*,*) 'gam2=',gam2
      write(*,*) 'rho2=',r2
      write(*,*) 'e2=',e2
      p2=p0                              !Pressure balance at interface.
      write(*,*) 'p2=',p2
      c2 = sqrt(gam2*p2/r2)              !Sound speed in Region 2.
      write(*,*) 'Sound speed in region 2, c2=',c2
      write(*,*)
      p1op0 = (2.0*gamma*M*M-gamma+1.0)/(gamma+1.0)   !p1/p0.
      r1or0 = (gamma + 1.0)*p1op0+gamma-1.0
      r1or0 = r1or0/((gamma-1.0)*p1op0+gamma+1.0)     !rho1/rho0.
      v1    = c0*2.0*(M*M-1.0)/(M*(gamma+1.0))        !Piston Velocity.
      e1oe0 = p1op0/r1or0                             !internal energy.
      write(*,*) 'Shock processed region 0 (i.e. region 1):'
      write(*,*)
      write(*,*) 'p1/p0=',p1op0
      write(*,*) 'rho1/rho0=',r1or0
      write(*,*) 'rho1=',r1or0*r0
      write(*,*) 'Piston Velocity, v1=',v1
      write(*,*) 'e1/e0=',e1oe0
      write(*,*) 'e1=',e1oe0*e0
      write(*,*) 'Pressure boundary condition, p1=',p1op0*p0
      p1=p1op0*p0                   !Pressure of Region 1.
      r1=r1or0*r0                   !Density of Region 1.
      c1=sqrt(gamma*p1/r1)          !Sound speed of Region 1.
      write(*,*) 'Sound speed in region 1, c1=',c1
      write(*,*)
      x(1)=0.9*v1      !Initial guess for the speed of the interface.
      x(2)=0.5*(p2+p1) !Initial guess for the interfacial pressure.
      x(3)=M           !Initial guess for the Mach # of the transmitted shock.
```

```
 x(4)=M           !Initial guess for the Mach # of the reflected shock.
 call newt(x,4,check)    !Solve coupled nonlinear equation set.
 if (check.eq..false.) write(*,*) 'Normal Return'
 if (check.eq..true.) write(*,*) 'Bad return--local minimum?'
 uc=x(1)          !Speed of material interface.
 pc=x(2)          !Pressure at material interface.
 M23=x(3)         !Mach # of transmitted shock.
 M14=x(4)         !Mach # of reflected shock.
 call funcv(4,x,f)
 write(*,*) 'Eqn. 1',f(1)
 write(*,*) 'Eqn. 2',f(2)
 write(*,*) 'Eqn. 3',f(3)
 write(*,*) 'Eqn. 4',f(4)
 r3=-2.0*(1.0-1.0/M23**2)/(gam2+1.0)/r2+1.0/r2
 r3=1.0/r3        !Density of Region 3.
 write(*,*)
 write(*,*) 'Pressure at interface, pc=',pc
 write(*,*) 'Density in region 3, rho3=',r3
 if (pc.le.0.0) then
   write(*,*)
   write(*,*) 'Negative or zero interface pressure!'
   write(*,*) 'Try a slightly different Mach number.'
   write(*,*) 'Or adjust error bounds/initial guess in'
   write(*,*) 'solution algorithm.'
   write(*,*) 'DO NOT TRUST THE FOLLOWING NUMBERS!'
   pc=-pc
 end if
 c3=sqrt(gam2*pc/r3)             !Sound speed of Region 3.
 M35=0.5*(gam2+1.0)*uc/c3
 M35=0.5*(M35+sqrt(M35**2+4.0))  !Mach # of wall reflected shock.
 u23=M23*c2                      !Speed of transmitted shock.
 u35=M35*c3-uc                   !Speed of wall reflected shock.
 write(*,*)
 write(*,*) 'Speeds of material interface and transmitted shock:'
 write(*,*)
 write(*,*) 'Speed of the interface, uc=',uc
 write(*,*) 'Speed of shock transmitted into region 2, u23=',u23
 write(*,*) 'Speed of shock reflected from wall, u35=',u35
 dum=(uc/u23+uc/u35)/(1.0+uc/u35)
 write(*,*) 'Collision point, x/L=',dum
 end

 SUBROUTINE newt(x,n,check)
 INTEGER n,nn,NP,MAXITS
 LOGICAL check
 REAL x(n),fvec,TOLF,TOLMIN,TOLX,STPMX
 PARAMETER (NP=40,MAXITS=200,TOLF=1.e-4,TOLMIN=1.e-6,TOLX=1.e-7,
*STPMX=100.)
 COMMON /newtv/ fvec(NP),nn
 SAVE /newtv/
 INTEGER i,its,j,indx(NP)
 REAL d,den,f,fold,stpmax,sum,temp,test,fjac(NP,NP),g(NP),p(NP),
*xold(NP),fmin
 EXTERNAL fmin
 nn=n
 f=fmin(x)
 test=0.
 do i=1,n
   if(abs(fvec(i)).gt.test)test=abs(fvec(i))
 end do
 if(test.lt..01*TOLF)then
   check=.false.
   return
 endif
 sum=0.
 do i=1,n
   sum=sum+x(i)**2
 end do
 stpmax=STPMX*max(sqrt(sum),float(n))
 do its=1,MAXITS
```

```
  call fdjac(n,x,fvec,NP,fjac)
  do i=1,n
    sum=0.
    do j=1,n
      sum=sum+fjac(j,i)*fvec(j)
    end do
    g(i)=sum
  end do
  do i=1,n
    xold(i)=x(i)
  end do
  fold=f
  do i=1,n
    p(i)=-fvec(i)
  end do
  call ludcmp(fjac,n,NP,indx,d)
  call lubksb(fjac,n,NP,indx,p)
  call lnsrch(n,xold,fold,g,p,x,f,stpmax,check,fmin)
  test=0.
  do i=1,n
    if(abs(fvec(i)).gt.test)test=abs(fvec(i))
  end do
  if(test.lt.TOLF)then
    check=.false.
    return
  endif
  if(check)then
    test=0.
    den=max(f,.5*n)
    do i=1,n
      temp=abs(g(i))*max(abs(x(i)),1.)/den
      if(temp.gt.test)test=temp
    end do
    if(test.lt.TOLMIN)then
      check=.true.
    else
      check=.false.
    endif
    return
  endif
  test=0.
  do i=1,n
    temp=(abs(x(i)-xold(i)))/max(abs(x(i)),1.)
    if(temp.gt.test)test=temp
  end do
  if(test.lt.TOLX)return
end do
pause 'MAXITS exceeded in newt'
end

SUBROUTINE fdjac(n,x,fvec,np,df)
INTEGER n,np,NMAX
REAL df(np,np),fvec(n),x(n),EPS
PARAMETER (NMAX=40,EPS=1.e-4)
INTEGER i,j
REAL h,temp,f(NMAX)
do j=1,n
  temp=x(j)
  h=EPS*abs(temp)
  if(h.eq.0.)h=EPS
  x(j)=temp+h
  h=x(j)-temp
  call funcv(n,x,f)
  x(j)=temp
  do i=1,n
    df(i,j)=(f(i)-fvec(i))/h
  end do
end do
return
end
```

```
      FUNCTION fmin(x)
      INTEGER n,NP
      REAL fmin,x(*),fvec
      PARAMETER (NP=40)
      COMMON /newtv/ fvec(NP),n
      SAVE /newtv/
      INTEGER i
      REAL sum
      call funcv(n,x,fvec)
      sum=0.0
      do i=1,n
        sum=sum+fvec(i)**2
      end do
      fmin=0.5*sum
      return
      end

      SUBROUTINE lnsrch(n,xold,fold,g,p,x,f,stpmax,check,func)
      INTEGER n
      LOGICAL check
      REAL f,fold,stpmax,g(n),p(n),x(n),xold(n),func,ALF,TOLX
      PARAMETER (ALF=1.e-4,TOLX=1.e-7)
      EXTERNAL func
      INTEGER i
      REAL a,alam,alam2,alamin,b,disc,f2,fold2,rhs1,rhs2,slope,sum,temp,
     *test,tmplam
      check=.false.
      sum=0.
      do i=1,n
        sum=sum+p(i)*p(i)
      end do
      sum=sqrt(sum)
      if(sum.gt.stpmax)then
        do i=1,n
          p(i)=p(i)*stpmax/sum
        end do
      endif
      slope=0.
      do i=1,n
        slope=slope+g(i)*p(i)
      end do
      test=0.
      do i=1,n
        temp=abs(p(i))/max(abs(xold(i)),1.)
        if(temp.gt.test)test=temp
      end do
      alamin=TOLX/test
      alam=1.
1     continue
        do i=1,n
          x(i)=xold(i)+alam*p(i)
        end do
        f=func(x)
        if(alam.lt.alamin)then
          do i=1,n
            x(i)=xold(i)
          end do
          check=.true.
          return
        else if(f.le.fold+ALF*alam*slope)then
          return
        else
          if(alam.eq.1.)then
            tmplam=-slope/(2.*(f-fold-slope))
          else
            rhs1=f-fold-alam*slope
            rhs2=f2-fold2-alam2*slope
            a=(rhs1/alam**2-rhs2/alam2**2)/(alam-alam2)
            b=(-alam2*rhs1/alam**2+alam*rhs2/alam2**2)/(alam-alam2)
```

```
      if(a.eq.0.)then
        tmplam=-slope/(2.*b)
      else
        disc=b*b-3.*a*slope
        if(disc.lt.0.) pause 'roundoff problem in lnsrch'
        tmplam=(-b+sqrt(disc))/(3.*a)
      endif
      if(tmplam.gt..5*alam)tmplam=.5*alam
    endif
  endif
  alam2=alam
  f2=f
  fold2=fold
  alam=max(tmplam,.1*alam)
goto 1
end

SUBROUTINE lubksb(a,n,np,indx,b)
INTEGER n,np,indx(n)
REAL a(np,np),b(n)
INTEGER i,ii,j,ll
REAL sum
ii=0
do i=1,n
  ll=indx(i)
  sum=b(ll)
  b(ll)=b(i)
  if (ii.ne.0)then
    do j=ii,i-1
      sum=sum-a(i,j)*b(j)
    end do
  else if (sum.ne.0.) then
    ii=i
  endif
  b(i)=sum
end do
do i=n,1,-1
  sum=b(i)
  do j=i+1,n
    sum=sum-a(i,j)*b(j)
  end do
  b(i)=sum/a(i,i)
end do
return
end

SUBROUTINE ludcmp(a,n,np,indx,d)
INTEGER n,np,indx(n),NMAX
REAL d,a(np,np),TINY
PARAMETER (NMAX=500,TINY=1.0e-20)
INTEGER i,imax,j,k
REAL aamax,dum,sum,vv(NMAX)
d=1.0
do i=1,n
  aamax=0.
  do j=1,n
    if (abs(a(i,j)).gt.aamax) aamax=abs(a(i,j))
  end do
  if (aamax.eq.0.) pause 'singular matrix in ludcmp'
  vv(i)=1./aamax
end do
do j=1,n
  do i=1,j-1
    sum=a(i,j)
    do k=1,i-1
      sum=sum-a(i,k)*a(k,j)
    end do
    a(i,j)=sum
  end do
  aamax=0.
```

```
  do i=j,n
    sum=a(i,j)
    do k=1,j-1
      sum=sum-a(i,k)*a(k,j)
    end do
    a(i,j)=sum
    dum=vv(i)*abs(sum)
    if (dum.ge.aamax) then
      imax=i
      aamax=dum
    endif
  end do
  if (j.ne.imax)then
    do k=1,n
      dum=a(imax,k)
      a(imax,k)=a(j,k)
      a(j,k)=dum
    end do
    d=-d
    vv(imax)=vv(j)
  endif
  indx(j)=imax
  if(a(j,j).eq.0.)a(j,j)=TINY
  if(j.ne.n)then
    dum=1./a(j,j)
    do i=j+1,n
      a(i,j)=a(i,j)*dum
    end do
  endif
end do
return
end

subroutine funcv(n,x,fvec)
integer n
real x(n),fvec(n)
real gamma,gam2,p2,p1,c1,c2,v1
common /pars/ gamma,gam2,p2,p1,c1,c2,v1
fvec(1)=x(1)/c2-2.0*(x(3)-1.0/x(3))/(gam2+1.0)
fvec(2)=(x(1)-v1)/c1+2.0*(x(4)-1.0/x(4))/(gamma+1.0)
fvec(3)=(x(2)-p2)/p2-2.0*gam2*(x(3)*x(3)-1.0)/(gam2+1.0)
fvec(4)=(x(2)-p1)/p1-2.0*gamma*(x(4)*x(4)-1.0)/(gamma+1.0)
return
end
```

---

[1] Y. B. Zel'dovich and Y. P. Raizer, *Physics of Shock Waves and High-Temperature Hydrodynamic Phenomena, Vol. I*, Academic Press, New York, 1966, p. 49.

[2] R. D. Richtmyer, Taylor instability in shock acceleration of compressible fluids, *Communications on Pure and Applied Mathematics*, **13**, 297, 1960.

[3] E. E. Meshkov, Instability of the interface of two gases accelerated by a shock wave, *Fluid Dynamics*, **4**, 101, 1969.

[4] H. W. Liepmann, A. Roshko, D. Coles, B. Sturtevant, A 17-inch diameter shock tube for studies in rarefied gas dynamics. *Rev. Sci. Instrum.*, **33**, 625, 1962.

[5] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in FORTRAN, The Art of Scientific Computing (2nd ed.)*, Cambridge University Press, New York, 1992, p. 376.